Terms used **object oriented dynamic link library** Found **86,794** of **141,680**

Sort results by relevance Save results to a Binder

Display results expanded form Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown Relevance scale

1 Portable run-time support for dynamic object-oriented parallel processing

Andrew S. Grimshaw, Jon B. Weissman, W. Timothy Strayer

May 1996 **ACM Transactions on Computer Systems (TOCS)**, Volume 14 Issue 2

Full text available: pdf(2.21 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Mentat is an object-oriented parallel processing system designed to simplify the task of writing portable parallel programs for parallel machines and workstation networks. The Mentat compiler and run-time system work together to automatically manage the communication and synchronization between objects. The run-time system marshals member function arguments, schedules objects on processors, and dynamically constructs and executes large-grain data dependence graphs. In this article we presen ...

Keywords: MIMD, dataflow, distributed memory, object-oriented, parallel processing

2 Special issue on persistent object systems: Orthogonally persistent object systems

Malcolm Atkinson, Ronald Morrison

July 1995 **The VLDB Journal — The International Journal on Very Large Data Bases**, Volume 4 Issue 3

Full text available: pdf(5.02 MB) Additional Information: full citation, abstract, references, citings

Persistent Application Systems (PASs) are of increasing social and economic importance. They have the potential to be long-lived, concurrently accessed, and consist of large bodies of data and programs. Typical examples of PASs are CAD/CAM systems, office automation, CASE tools, software engineering environments, and patient-care support systems in hospitals. Orthogonally persistent object systems are intended to provide improved support for the design, construction, maintenance, and operation o ...

Keywords: database programming languages, orthogonal persistence, persistent application systems, persistent programming languages

3 Type-Safe linking with recursive DLLs and shared libraries

Dominic Duggan

November 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 6

Full text available: pdf(658.62 KB) Additional Information: full citation, abstract, references, citings, index

terms

Component-based programming is an increasingly prevalent theme in software development, motivating the need for expressive and safe module interconnection languages. Dynamic linking is an important requirement for module interconnection languages, as exemplified by dynamic link libraries (DLLs) and class loaders in operating systems and Java, respectively. A semantics is given for a type-safe module interconnection language that supports shared libraries and dynamic linking, as well as circular ...

Keywords: Dynamic Linking, Module Interconnection Languages, Recursive Modules, Shared Libraries

4 Extracting library-based object-oriented applications

Peter F. Sweeney, Frank Tip

November 2000 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 8th ACM SIGSOFT international symposium on Foundations of software engineering: twenty-first century applications**, Volume 25 Issue 6

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, citings, index terms

In an increasingly popular model of software distribution, software is developed in one computing environment and deployed in other environments by transfer over the internet. Extraction tools perform a static whole-program analysis to determine unused functionality in applications in order to reduce the time required to download applications. We have identified a number of scenarios where extraction tools require information beyond what can be inferred through static analysis: software distr ...

5 Object-oriented programming languages and systems (OOP): Modeling multiple class loaders by a calculus for dynamic linking

Sonia Fagorzi, Elena Zucca, Davide Ancona

March 2004 **Proceedings of the 2004 ACM symposium on Applied computing**

Full text available: pdf(243.15 KB) Additional Information: full citation, abstract, references

In a recent paper we proposed a calculus for modeling dynamic linking independently of the details of a particular programming environment.Here we use a particular instantiation of this calculus to encode a toy language, called MCL, which provides an abstract view of the mechanism of dynamic class loading with multiple loaders as in Java.The aim is twofold. On one hand, we show an example of application of the calculus in modeling existing loading and linking policies, showing in particular that ...

Keywords: Java, dynamic linking, multiple loaders

6 The impact of Ada and object-oriented design in NASA Goddard's Flight Dynamics Division

Sharon Waligora, John Bailey, Mike Stark

May 1997 **ACM SIGAda Ada Letters**, Volume XVII Issue 3

Full text available: pdf(1.68 MB) Additional Information: full citation, abstract, index terms

This paper presents the highlights and key findings of 10 years of use and study of Ada and object-oriented design in NASA Goddard's Flight Dynamics Division (FDD). In 1985, the Software Engineering Laboratory (SEL) began investigating how the Ada language might apply to FDD software development projects. Although they began cautiously using Ada on only a few pilot projects, they expected that, if the Ada pilots showed promising results, the FDD would fully transition its entire development orga ...

7 Product metrics for object-oriented systems

Sandeep Purao, Vijay Vaishnavi

June 2003 **ACM Computing Surveys (CSUR)**, Volume 35 Issue 2

Full text available: pdf(363.99 KB) Additional Information: full citation, abstract, references, index terms

We survey metrics proposed for object-oriented systems, focusing on product metrics. The survey is intended for the purposes of understanding, classifying, and analyzing ongoing research in object-oriented metrics. The survey applies fundamental measurement theory to artifacts created by development activities. We develop a mathematical formalism that captures this perspective clearly, giving appropriate attention to the peculiarities of the object-oriented system developmenr process. Consistent ...

Keywords: Software metrics, measurement theory, object-oriented metrics, object-oriented product metrics, object-oriented systems

8 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

9 More dynamic object reclassification: *Fickle*∥

Sophia Drossopoulou, Ferruccio Damiani, Mariangiola Dezani-Ciancaglini, Paola Giannini

March 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 2

Full text available: pdf(365.61 KB) Additional Information: full citation, abstract, references, citings, index terms

Reclassification changes the class membership of an object at run-time while retaining its identity. We suggest language features for object reclassification, which extend an imperative, typed, class-based, object-oriented language.We present our proposal through the language *Fickle*⋄⋄. The imperative features, combined with the requirement for a static and safe type system, provided the main challenges. We develop a type and effect system for *Fickle* ...

Keywords: Object-oriented languages, type and effect systems

10 Reconciling responsiveness with performance in pure object-oriented languages

Urs Hölzle, David Ungar

July 1996 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 18 Issue 4

Full text available: pdf(537.19 KB) Additional Information: full citation, abstract, references, citings, index terms, review

Dynamically dispatched calls often limit the performance of object-oriented programs, since opject-oriented programming encourages factoring code into small, reusable units, thereby increasing the frequency of these expensive operations. Frequent calls not only slow down execution with the dispatch overhead per se, but more importantly they hinder optimization by limiting the range and effectiveness of standard global optimizations. In particular,

dynamically dispatched calles prevent stand ...

Keywords: adaptive optimization, pause clustering, profile-based optimization, run-time compilation, type feedback

11 Computational mechanics solvers based on object-oriented design principles

Joseph Whitesell, John D. Reid

December 1990 **Proceedings of the 22nd conference on Winter simulation**

Full text available: pdf(499.94 KB) Additional Information: full citation, references, index terms

12 A hierarchy-aware approach to faceted classification of objected-oriented components

E. Damiani, M. G. Fugini, C. Bellettini

July 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 8 Issue 3

Full text available: pdf(310.25 KB) Additional Information: full citation, abstract, references, citings, index terms, review

This article presents a hierarchy-aware classification schema for obje ct-oriented code, where software components are classified according to their behavioral characteristics, such as provided services, employed algorithms, and needed data. In the case of reusable application frameworks, these characteristics are constructured from their model, i.e., from the description of the abstract classes specifying both the framework structure and purpose. In conven ...

Keywords: code analysis, component repositories, component retrieval, software reuse, user feedback

13 Type-based hot swapping of running modules (extended abstract)

Dominic Duggan

October 2001 **ACM SIGPLAN Notices , Proceedings of the sixth ACM SIGPLAN international conference on Functional programming**, Volume 36 Issue 10

Full text available: pdf(150.34 KB) Additional Information: full citation, abstract, references, citings, index terms

While dynamic linking has become an integral part of the run-time execution of modem programming languages, there is increasing recognition of the need for support for hot swapping of running modules, particularly in long-lived server applications. The interesting challenge for such a facility is to allow the new module to change the types exported by the original module, while preserving type safety. This paper describes a type-based approach to hot swapping running modules. The approach is bas ...

Keywords: dynamic typing, hot swapping, module interconnection languages, shared libraries

14 Corrigenda: a hierarchy-aware approach to faceted classification of object-oriented components

E. Damiani, M. G. Fugini, C. Bellettini

October 1999 **ACM Transactions on Software Engineering and Methodology (TOSEM)**, Volume 8 Issue 4

Full text available: pdf(310.50 KB) Additional Information: full citation, abstract, references, index terms

This article presents a hierarchy-aware classification schema for object-oriented code, where

software components are classified according to their behavioral characteristics, such as provided services, employed algorithms, and needed data. In the case of reusable application frameworks, these characteristics are constructed from their model, i.e., from the description of the abstract classes specifying both the framework structure and purpose. In conventio ...

15 Object-oriented composition untangled

Klaus Ostermann, Mira Mezini

October 2001 **ACM SIGPLAN Notices , Proceedings of the 16th ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications**, Volume 36 Issue 11

Full text available: pdf(681.61 KB) Additional Information: full citation, abstract, references, citings, index terms

Object-oriented languages come with pre-defined composition mechansims, such as inheritance, object composition, or delegation, each characterized by a certain set of composition properties, which do not themselves individually exist as abstractions at the language level. However, often non-standard composition semantics is needed, with a mixture of composition mechanisms. Such non-standard semantics are simulated by complicated architectures that are sensitive to requirement changes and cannot ...

16 The object-oriented paradigm and neurocomputing

Paul S. Prueitt, Robert M. Craig

May 1991 **Proceedings of the conference on Analysis of neural network applications**

Full text available: pdf(1.65 MB) Additional Information: full citation, references, index terms

17 HydroJ: object-oriented pattern matching for evolvable distributed systems

Keunwoo Lee, Anthony LaMarca, Craig Chambers

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th ACM SIGPLAN conference on Object-oriented programing, systems, languages, and applications**, Volume 38 Issue 11

Full text available: pdf(311.06 KB) Additional Information: full citation, abstract, references, index terms

In an evolving software system, components must be able to change independently while remaining compatible with their peers. One obstacle to independent evolution is the *brittle parameter problem*: the ability of two components to communicate can depend on a number of *inessential* details of the types, structure, and/or contents of the values communicated. If these details change, then the components can no longer communicate, even if the *essential* parts of the message remain ...

Keywords: HydroJ, XML, distributed systems, dynamic dispatch, object-oriented programming, pattern matching, semi-structured data, software evolution, ubiquitous computing

18 Understanding object-oriented: a unifying paradigm

Tim Korson, John D. McGregor

September 1990 **Communications of the ACM**, Volume 33 Issue 9

Full text available: pdf(2.30 MB) Additional Information: full citation, abstract, references, citings, index terms, review

The need to develop and maintain large complex software systems in a competitive and dynamic environment has driven interest in new approaches to software design and development. The problems with the classical waterfall model have been cataloged in almost every software engineering text [19,23]. In response, alternative models such as the spiral

[2], and fountain [9] have been proposed. Problems with traditional development using the classical life cycle include no iteration, no ...

19 On type systems for object-oriented database programming languages

Yuri Leontiev, M. Tamer Özsu, Duane Szafron

December 2002 **ACM Computing Surveys (CSUR)**, Volume 34 Issue 4

Full text available: pdf(346.87 KB) Additional Information: full citation, abstract, references, index terms

The concept of an object-oriented database programming language (OODBPL) is appealing because it has the potential of combining the advantages of object orientation and database programming to yield a powerful and universal programming language design. A uniform and consistent combination of object orientation and database programming, however, is not straightforward. Since one of the main components of an object-oriented programming language is its type system, one of the first problems that ar ...

Keywords: OODB, OODBPL, object-oriented database programming language, type checking, typing

20 Roles for composite objects in object-oriented analysis and design

Franco Civello

October 1993 **ACM SIGPLAN Notices , Proceedings of the eighth annual conference on Object-oriented programming systems, languages, and applications**, Volume 28 Issue 10

Full text available: pdf(1.86 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next